

Team member: Bingxu Hu (bhu62@wisc.edu) Xuanhan Liu (xliu2238@wisc.edu) Pu Guo (pguo25@wisc.edu)

Introduction:

This is an E-Commerce Application, users are allowed to sell their goods or purchase items they want. For the market seller, they can release items to their market shop page and edit the item details like the item name, price, and description. For Customers, they can search the keyword to find the relevant items and purchase the order to get items.

Entity:

1. Customer: Entity that purchases in the application.
2. Order: Entity that represents the transaction for specific items.
3. Market Seller: Entity that sells items on the application to customers.
4. Delivery: Weak entity of Order, which tracks the shipment status for every item in the order.
5. Item: Entity provided by sellers to be shopped by customers.
6. Virtual Item: Subclass of items representing virtual things such as game accounts, gift code, and software. We use the ER approach to handle this subclass.

Relation:

1. Customer - Order -> One to Many: One customer could make many orders in this application. We merge the relation to Order, order have customer id as attribute.
2. Item - Order -> Many to Many: An order could include many items for the customer to pay at once, an item could be included in many orders. We keep a single table for this relation to keep it safe.
3. Item - Market Seller -> Many to One: One seller could sell different kinds of items in their shop. We merge this relation to Item.
4. Order - Delivery -> One to many: Each order has delivery which tracks all items.
5. Item- Delivery -> Many to one: One Item could be in many orders and could be tracked by many delivery.
6. Customer - Contact_Info: One to One: Each Customer have only one contact_info and one contact_info belongs to only one customer

Frequent query:

1. SELECT * FROM Order WHERE Order.customer = 123 (get all orders made by customer 123)
2. SELECT Item.*, Order_Item.item_quantity FROM Item, Order_Item WHERE order_id = 1234 AND Order_Item.item_id = item.item_id (get all item tuple with its quantity in the order 1234)
3. SELECT Item.name, Delivery.d_status FROM Item, Delivery, Order_Item WHERE Delivery.order_id = Order_Item.order_id AND Delivery.Item_id = Item.Item_id (track shipment status for every item in the order)
4. SELECT * FROM Item WHERE shop_id = 123 (get all Item tuple provided by shop 123)
5. SELECT * FROM Contact_Info WHERE customer_id = 123 (get all contact info of customer 123)



